Configurable AI Framework for Multimodal News Content Generation

Zachary Wiel | San Francisco, CA
zwiel@tulane.edu

Alex Olteanu | Queens, NY
aolteanu1@tulane.edu

Abstract

Configurable Artificial Intelligence Framework for Multimodal News Content Generation (CFMN-CG) aims to tackle the difficulty and time consumption of staying informed on financial markets. Our proprietary technology is meant to bridge the divide between institutional investors and retail traders, hoping to empower "the little guy". In its current state, our program can distill information related to requested stock tickers. Eventually, the scope of our project will be scaled to include general market current events, general world and political events, and finally, any educational topic desired by the user. Traditional means of consuming financial information and current events can be dense and jargon-heavy. Our solution tackles today's "attention economy" by providing an accessible and entertaining way to stay informed.

Introduction

Configurable Artificial Intelligence Framework for Multimodal News Content Generation (CFMN-CG) provides a customizable framework for generating information content in various media formats. This is achieved by using a Python framework that processes financial information by taking user requests for stock tickers and media formats (text, video, or audio). It scrapes Yahoo Finance articles, analyzes the content using GPT, and formats the output accordingly. Text results are saved as files, while audio and video content are created using Whisper TTS and Stability AI's animation tools. The framework also indexes data for query handling and is accessible through both a web application and a command line interface.

Related Work

Currently, there is limited evidence to suggest that existing projects have directly tackled the comprehensive functionality described in this paper. Instead, there exist various products and technologies that specialize in addressing specific subsets of the functionalities we aimed to achieve.

Methods

Our Python framework begins by capturing user input through a Request class, which leverages GPT functions to identify the requested stock (via its ticker) and the desired media format (text, video, or audio). This stock ticker information is then relayed to a WebScraper class, which focuses on scraping Yahoo Finance articles related to the specified ticker, compiling the gathered data into a JSON document.

The scraped articles are processed by the Model class, where a GPT-based analysis interprets the information. The output is formatted according to the media type determined earlier—text, audio, or video. For video requests, an additional GPT call generates a prompt for an animation model to ensure the content is visually engaging.

The final output is managed by the Media class: text outputs are saved into a text file, while audio and video outputs are handled through OpenAI's Whisper TTS model to generate narration. For video content, Stability AI's stability-sdk API is employed to create corresponding animations, which are synced with the audio to produce a complete video.

Moreover, all collected research is indexed into a vector database using Llama Index, enhancing our framework's ability to answer queries about the stock through a Conversation class. This entire pipeline is accessible via a Flask web application and a command-line interface, providing versatile user interaction options.

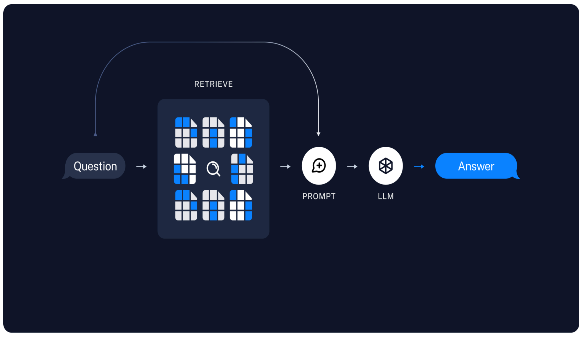
Retrieval Augmented Generation (RAG)

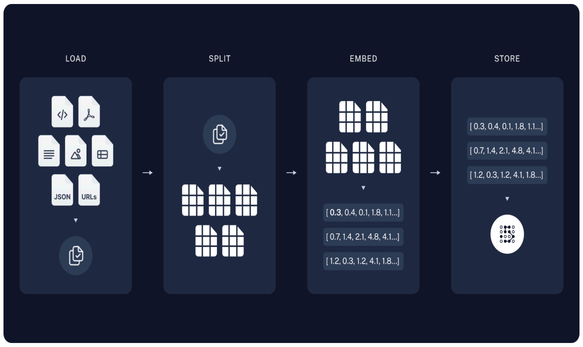
Retrieval Augmented Generation or RAG, is a technique used to enhance the capabilities of Large Language Models (LLMs) by incorporating user-specific data in their responses. While LLMs are trained on vast amounts of data, they lack direct access to user-specific or organization-level information. RAG addresses this dilemma by integrating user data into LLM prompts, allowing them to provide more personalized and contextually relevant responses.

The RAG process involves several key stages. These are loading, indexing, storing, querying, and evaluation. The loading stage is where data is gathered from diverse sources such as text files, databases, PDFs, APIs, or websites. This data is then prepared for processing by creating data structures that allow for readily querying the data. In the context of LLMs, this often means creating vector embeddings, or numerical representations of the user’s data.

Once the data has been indexed, it is typically stored in a vector database. This streamlines the retrieval process. Next, the querying stage follows. User queries and processed and matched against the indexed data. RAG aims to identify and extract the most relevant information based on the user’s input, ensuring that the responses and appropriate and contextually relevant. Lastly, evaluation must be performed on behalf of the developers, where system performance is assessed.

In practice, RAG has applications across a wide range of use cases such as customer support chatbots, information distillation, content creation, and many more. The versatility of RAG makes it a valuable tool for empowering LLMs to deliver personalized and contextually relevant interactions.





Results

In its current state, our project enables users to request information related to a stock ticker. While we initially aimed to incorporate natural language processing for user requests, we encountered unexpected challenges that we are actively addressing. Nonetheless, users can specify their desired stock ticker and preferred media format, allowing the generation of relevant content. Following the generation of media, users can engage further by asking questions related to the content they have just viewed. Despite the encountered obstacles, we were able to deliver a decent portion of the functionality we envisioned.

Division of Labor

In our project, the division of labor was structured to leverage the experience of each group member. Zach primarily focused on building the object-oriented programming framework and developing general classes needed for the project's functionality. Meanwhile, Alex took the lead in implementing the web scraping component and spearheaded the integration of the Retrieval-Augmented Generation (RAG) techniques

Communication played a crucial role in our collaboration, with regular discussions and brainstorming sessions enabling us to align our efforts effectively and make informed decisions. Overall, the division of labor was equitable, with tasks tackled as they appeared.

Conclusion

In conclusion, the Configurable Artificial Intelligence Framework for Multimodal News Content Generation (CFMN-CG) has not only met many of our initial goals but has also highlighted a promising path forward for further innovation and development. Our framework successfully enables users to access and understand financial information through a user-friendly multimodal approach, which is particularly beneficial in today's fast-paced, information-heavy financial environments. Moving forward, we plan to enhance our framework by incorporating more advanced language models and expanding the range of media formats supported. This includes a focus on improving the real-time responsiveness of our system to financial news and integrating more comprehensive data analysis features.

Additionally, we aim to refine our web scraper to handle a broader spectrum of content requests and to support a wider variety of financial instruments. These enhancements will help ensure that our framework remains both current and highly relevant to the needs of users ranging from retail investors to professional traders.

Beyond technical improvements, we also see significant potential in exploring the broader implications of our technology. By making financial information more accessible and easier to digest, CFMN-CG contributes to financial literacy and empowers individual investors, thereby democratizing access to information that was traditionally dominated by institutional entities. Furthermore, the principles and technology underlying our framework have the potential to be adapted for educational purposes across different fields, helping to simplify complex information and make learning more engaging and effective.

As we continue to refine CFMN-CG, our focus will remain on enhancing its functionality and reach, ensuring that it not only serves as a tool for financial information but also as a platform for broader educational and informational empowerment. The journey thus far has been both challenging and rewarding, and we are eager to face the opportunities and challenges that lie ahead.

References

https://bgr.com/business/ai-powered-search-engine-perplexity-launches-an-ai-generated-daily-news-podcast/

https://aicontentfy.com/en/blog/ai-generated-podcasts-new-way-to-listen

https://aicontentfy.com/en/blog/ai-generated-content-for-podcasting-and-audio-content

https://aicontentfy.com/en/blog/ai-generated-podcasts-changing-game

https://aicontentfy.com/en/blog/chatgpt-and-future-of-ai-generated-content-for-podcasts

https://speechify.com/blog/how-to-make-ai-podcasts/

https://www.thepodcasthost.com/planning/ai-podcasting-tools/

https://rss.com/blog/the-best-ai-tools-for-podcasting-and-how-to-use-them/

https://platform.openai.com/docs/guides/text-to-speech

https://github.com/FujiwaraChoki/MoneyPrinter

https://github.com/RayVentura/ShortGPT

https://docs.llamaindex.ai/en/stable/getting_started/concepts/

https://python.langchain.com/docs/use_cases/question_answering/